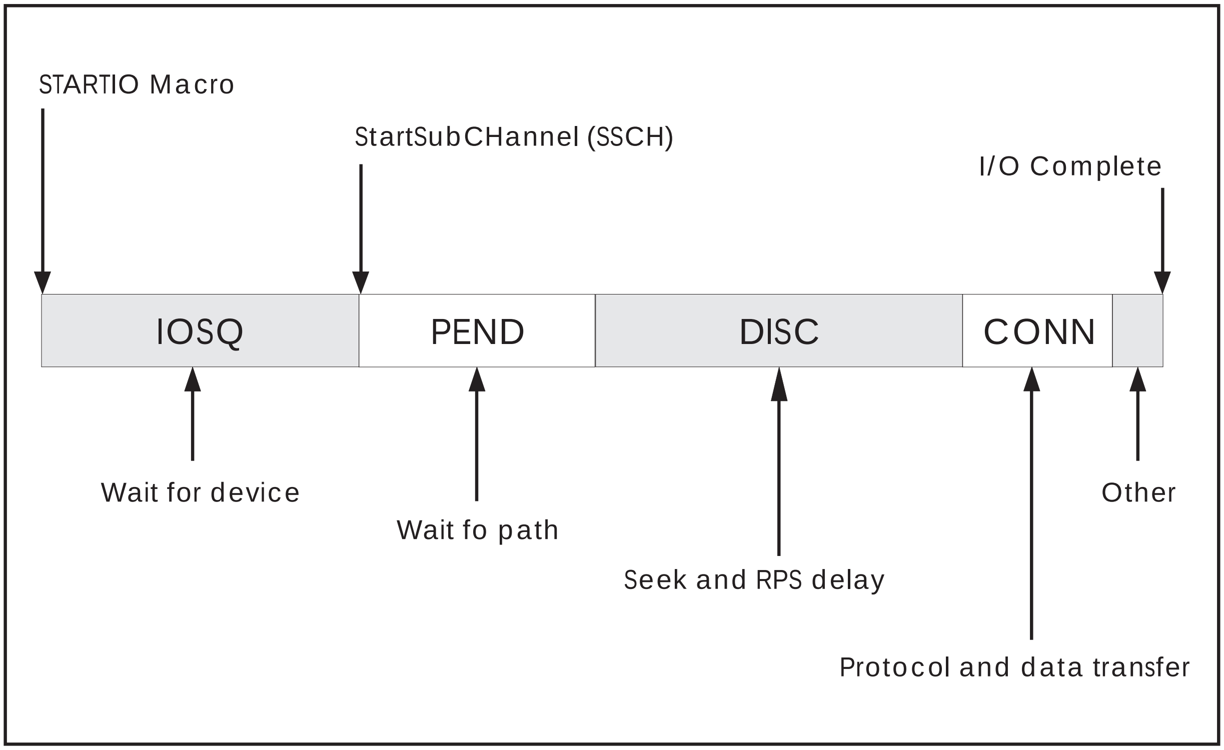
Section 5: DASD Analysis Factors

This section discusses DASD performance analysis considerations. Chapter 1 presents an overview of performance factors from a DASD I/O operation viewpoint. Chapter 2 highlights some of the factors that must be considered when analyzing DASD performance based upon data collected and recorded by SMF or RMF.

Chapter 1: Overview of DASD Performance Considerations

From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. These measures are reported by RMF in SMF Type 74 records. Exhibit 5-1 illustrates these four measures and another potential element of DASD I/O time, titled "Other".



MAJOR COMPONENTS OF DASD I/O OPERATIONS

EXHIBIT 5-1

Chapter 1.1: IOSQ time

IOSQ time is the time from the issuance of a STARTIO macro until the Start SubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the device is busy. If the device is not busy with this system, the SSCH instruction is issued. However, if the device is busy with this system, the I/O request is queued.

Thus, IOSQ time always means that the device is unable to handle additional requests from this system. (The emphasis on "this system" is explained in the below discussion of PEND time.)

Some small IOSQ time is often unavoidable. However, large IOSQ time imply a situation that should be examined. Large IOSQ times result from (1) too many I/O operations directed to the device or (2) lengthy device response times (perhaps caused by high seeking, high RPS delays, or high PEND time). Large IOSQ times usually involve the following situations:

- Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing for the single volume.

- Multiple users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the IOSQ problems.

- Multiple application systems may be using the volume experiencing high IOSQ times. In this case, perhaps application redesign or scheduling can solve the problem.

- A particular application (or system function) may be executing I/O to the device faster than the device can respond.

- The overall device response time (PEND, DISC, and CONN) times may be large, such that the device is unable to provide quick response to the I/O requests. This situation will be revealed by large values in the PEND, DISC, or CONN measures.

With Parallel Access Volumes (PAV) and dynamic alias management in Goal Mode, IOSQ time should be significantly reduced or eliminated. The implications of PAV and dynamic alias management will be discussed in rules related to these features.

Chapter 1.2: PEND time

PEND time is the time from the issuance of the SSCH instruction until the device is selected by the control unit. This time is caused by queuing for the path (wait for channel, wait for control unit or wait for head-of-string), and can be caused by other systems sharing the device (wait for device). Large PEND times usually involve the following situations:

- **Shared devices**. If the device is shared with another system, PEND time may indicate contention with the other system. Large PEND times in shared-device environments usually involve situations very similar to those described under IOSQ time:

  - Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing at the channel level (reflected as PEND time) for the single volume.

    If some of the data sets are not required to be shared, then the Data Base Administrator has complete flexibility to move these data sets (subject, of course, to the performance implications of the target devices). These data sets should be moved to a non-shared device.

    If the data sets are required to be shared, then they must be relocated to shared devices.

  - Multiple applications or users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set may be placed on different volumes. This would solve the PEND problems cause by contending systems. If this option is feasible, the data sets could be placed on non-shared devices, likely resulting in even more performance improvement.

  - Multiple application systems may be using the volume experiencing high PEND times. In this case, perhaps application redesign or scheduling can solve the problem.

  Additionally, large PEND times for shared devices could be caused by RESERVE from the other system. The applications issuing the RESERVE should be examined to determine whether the RESERVE is required. If the RESERVE **is** required, the above situations should reviewed to determine whether improvements can be achieved.

- **Non-shared devices**. Large PEND times for devices that are not shared may mean that there are insufficient paths available to the device. Too much I/O may be directed to many devices on the path, control unit, or head-of-string. The data sets can be redistributed among different volumes on different paths, control units, or heads-of-string. This will reduce the hardware-level queuing. Alternatively, the entire volume may be moved to a different (less busy) head-of-string or path.

  If redistributing the data sets or moving the volume is not feasible, then the device should have more paths. Depending upon the existing configuration, this may involve re-configuring existing channel paths, or acquiring additional hardware.

Fortunately, SMF Type 78 and Type 74 records contain information which can be used to identify at which level the hardware queuing occurs (that is, whether queuing is for the path, control unit, or head-of-string).

- **Devices attached to cached controllers**. Large PEND times for devices attached to cached controllers may imply a high percent of read miss operations, or non-volatile storage (NVS) writes for IBM-3990-3 devices. Fairchild [1] lists four ways in which staging in caching controllers can cause hidden device busy (with the device busy potentially reflected in high PEND time):

  - The normal (random) caching algorithm stages all records to the end of the track after a requested record is read.

  - The normal (random) caching algorithm stages all records from the beginning of the track to the requested record if a front-end miss occurs.

  - The sequential caching algorithm stages all records to the end of the track after the requested record is read, and stages in all of the next track. IBM-3990 (Model 3) controllers stages in all of the next three tracks.

  - Most writes to extended function IBM-3990 (Model 3) go into NVS with a subsequent destaging required.

- **Dual Copy Initialize**. Large PEND times for IBM-3390 devices may be caused by dual copy initialize. In this case, the dual copy initialize should be turned off.

Chapter 1.3: DISC time

DISC time is the time (1) from when the controller initiates a SEEK Channel Command Word (and the seek requires an arm movement) on the device until the SEEK command is complete, (2) plus the time of the rotational delay while the SET SECTOR Channel Command Word is executing, and (3) plus the rotational position sensing (RPS) delay time required because of missed RPS reconnect.

- **Seek delay**. The SEEK command is responsible for positioning the arm to the proper cylinder. If no positioning is required (that is, the arm is already at the proper cylinder), the device is not disconnected. Seek operations occur because of accessing patterns with data sets and because of accessing patterns between data sets. Large seek times usually involve the following situations:

[1] Fairchild, Bill, "The Anatomy of an I/O Request", *Conference Proceedings*, CMG'90, the Computer Measurement Group, Chicago, IL.

- Multiple data sets being active on the volume. The data sets can be redistributed among different volumes, to eliminate the seeking on the single volume.

- Multiple users using the same data set on the volume. While only one data set is involved, the user or application accessing patterns may require frequent arm movement. A partitioned data set in which several TSO users reference different members is a common situation.

  Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the seeking problems.

- **Rotational delay**. The SET SECTOR command is responsible for locating the proper sector on the track as the disk rotates. (Actually, the SET SECTOR command locates a sector three sectors preceding the desired sector. This sector is called the *angular sector.*) The device is disconnected during the SET SECTOR command operation.

  The rotational delay may be from zero, to the total time required to rotate the disk to the required sector. It is possible that the required sector will be immediately under the head. In this case there is zero rotational delay. On the other hand, the sector could have just passed under the head before the SET SECTOR command was received by the drive. In this case, a full rotation must be accomplished before the required sector is located. On average, one-half of the rotation time will be required to locate the sector. This time is referred to as the **average latency** of the device. For example, IBM-3380 devices rotate every 16.6 milliseconds and the average latency is 8.3 milliseconds.

  It is important to realize that the latency is an average based upon many SET SECTOR commands. Any particular SET SECTOR command may have a latency ranging from zero to the maximum rotational delay.

  If there are few I/O commands for a particular device in a given measurement interval, it is uncertain what the average latency will be. However, if there are many I/O commands for a particular device in a given measurement interval, the average latency will normally be one-half of the rotational delay.

  The average latency may be (and should be) quite small with cached devices. This is because many I/O requests should be satisfied from the cache and have no latency.

- **Missed RPS reconnect**. The device attempts to reconnect to the path when the angular sector is reached (the angular sector is described above). If the reconnect attempt is successful before the desired sector is reached, then the device connects and the read or write operation can proceed. If the reconnect is not successful before the desired sector is reached, then the device does not connect, and a complete revolution of the track must occur before the angular sector is again reached. This is called a *missed rotational position sensing reconnect (or missed RPS reconnect)* delay.

There is no action which can alleviate the initial rotational positioning delay (aside from changing device characteristics, such as implementing caching or buffering at the device level). Over a large number of I/O operations, this initial delay will be one-half the rotation times.

However, the missed RPS delay is a function of the probability that the path will be busy when the device attempts to reconnect; the busier the path(s), the more missed RPS delay. (Note that the path busy time is a function of the connect time of other actuators. The path cannot be busy from the device itself when the device attempts to reconnect.)

There are no measurements available to determine precisely how the DISC time should be divided. However, queuing models can estimate the RPS reconnect delay time and assumptions can be made about the average latency time. By subtracting RPS reconnect delay and the average latency time from the DISC time, the result will be an approximation of the time spent seeking. The results from this method is valid if (1) the distribution of device access is random so that the queuing formulae are applicable and (2) a large number of I/O operations are involved so that the average latency assumption is proper.

Chapter 1.4: CONN time

CONN time is the time in which the device is actually connected to the path. This time includes the data transfer time, but also includes protocol exchange[2] (or "hand shaking") between the various components at several stages of the I/O operation.

The data transfer time obviously is a function of the amount of data being transferred. This simply is the number of bytes transferred divided by the transfer speed (for example, if 4096 bytes were transferred from an IBM-3380 with a transfer speed of 3,000,000 bytes per second, the 4096 bytes would require 4096/3,000,000 seconds; or about 1.36 milliseconds).

Large connect times generally are caused by the following situations:

- A large average block size. This situation may be highly desirable for sequential data sets, but would be quite undesirable for randomly accessed data.

- Long multi-track searches. For example, the catalog must be searched for cataloged files, the Volume Table of Contents (VTOC) must searched to find a requested file, a directory must be searched for partitioned data sets, etc.. These searches will result in long connect times for the volume involved.

[2] Note that the protocol exchange occurs at multiple points in the normal I/O operation, even though it is shown only once in Exhibit 5-1.

- Fairchild indicates that wrong sector numbers, non-IBM temporary error recovery, and coding OPTCD=W can also cause long connect times.

There have been no published "official" times for the protocol connect times. The protocol time has been estimated by different authors to range from .5 milliseconds to 1.5 milliseconds, depending upon the device and the authors' experiences. In any event, there generally is little that can be done about the protocol time.

Chapter 1.5: OTHER time

There are at least two other potential I/O delays for DASD: (1) waiting for the I/O completion interrupt to be serviced by a processor and (2) waiting for the I/O interrupt to be serviced by a domain under PR/SM. Neither potential I/O delay is expected to be of the magnitude of the four "standard" I/O delays. However, they can be significant in special circumstances.

- Multi-processor configurations running under MVS can use any processor to service an I/O interrupt. However, when a processor services an I/O interrupt, the processor's high-speed cache storage is no longer valid when control is returned to the interrupted task. Consequently, many of the processor's high-performance design features may be nullified.

  A hardware feature allows processors to be disabled for I/O interrupts. With this method, only a small number (perhaps only one) processor is enabled for interrupt processing. Only this processor will have its high-speed cache storage disturbed by the task-switching required for interrupt processing, and only this processor will periodically have its high-performance design features nullified. The disadvantage to this approach is that an interrupt may occur while the processor is busy servicing a previous interrupt.

  If an interrupt is pending and no processor is enabled to service the interrupt, the interrupt must wait until a processor is available. This time should be insignificant, unless the system is processing a significantly large number of I/O operations. If the system is processing a large number of I/O operations, the interrupt pending delay could pose performance problems.

  After the processor completes processing for an I/O interrupt, it issues a Test Pending Interrupt (TPI) instruction to determine whether there are any interrupts pending. If an I/O interrupt is pending, the processor proceeds to service that interrupt.

  The IEAOPTxx member of SYS1.PARMLIB contains the **CPENABLE** keyword. This keyword specifies the percent of I/O interrupts detected by the TPI instruction, compared with all I/O interrupts. When the percent exceeds the high threshold of the CPENABLE keyword, MVS enables another processor to handle pending I/O interrupts. If the percent falls below the low threshold of the CPENABLE keyword, MVS will disable a processor (to the point that only one processor is enabled). The low and high threshold values for CPENABLE are 10 and 30 percent, respectively. These values normally mean that less than 30% of the I/O interrupts will be delayed for I/O interrupt service.

- MVS environments running under as a guest under VM or in a logical partition (LPAR) under PR/SM are subject to I/O interrupt delays. These delays can occur if another guest (for VM) or another domain is in its dispatch interval when the I/O interrupt completion is posted. The I/O interrupt remains pending until the guest or domain is dispatched. These delays have been estimated to be far more significant than might otherwise be expected.

Neither of the potential I/O delays described above is measured by RMF (although RMF does provide information on the number of I/O interrupts serviced by each processor and the number of TPI instructions resulting in I/O interrupt servicing).

The potential I/O delays are included in this discussion of general DASD performance considerations because (1) they may become important under certain situations and (2) techniques may be developed to assess their impact.

Chapter 2: RMF Data Analysis Considerations

This chapter highlights some of the factors that must be considered when analyzing DASD information collected and recorded by SMF in Type 30 records or by RMF in Type 70(series) records.

These factors do **not** preclude a comprehensive analysis of performance data and usually do not prevent insight into the causes of unacceptable performance. However, the factors must be recognized and accounted for both by CPExpert in analyzing data and by the user in reviewing CPExpert's results. The factors stem from (1) the way in which SMF and RMF create and record Type 30 and Type 70(series) information, and (2) inherent limitations caused by data averages.

Chapter 2.1: SMF information

SMF Type 30 records contain a record sub-type code to identify when the records are written:

CODE	SUB-TYPE DESCRIPTION
1	Job start
2	Interval records
3	Step termination
4	Step total
5	Job termination
6	System address space

SMF will optionally record the different sub-types, depending upon parameters contained in the SMFPRMxx member of SYS1.PARMLIB. Most installations collect Sub-type 4 (Step total) records, and many installations collect Sub-type 2 (Interval) records. If Interval records are recorded by SMF, Sub-type 3 (Step termination) records are automatically created.

It is highly desirable to collect Interval/Step termination information. It is virtually impossible to analyze system performance based upon Step total information if there exists long-running jobs. This is because it is impossible to correlate the information reflected in the Step total records with the information contained in SMF Type 70(series) data.

The Sub-type 4 records are written only after a long-running job step terminates, while the SMF Type 70(series) records are written at user-defined intervals (the interval typically is every 30 minutes or so). Long-running job steps may span many RMF recording intervals [RMF is responsible for creating the SMF Type 70(series) records]. Consequently, there may be many RMF interval records written between the start and end of a long-running job step.

Sub-type 2 (Interval) records are written at user-defined intervals (typically the interval selected is the same interval as the RMF interval records). SMF writes a Sub-type 2 record when the specified interval has lapsed after the start of the job step and continues to write Sub-type 2 records at each subsequent interval. When the job step terminates, SMF writes a Sub-type 3 record containing the information since the last Sub-type 2 record was written. One consequence of the interval records is that system usage can be identified by workload, and can be correlated with the overall system statistics recorded by RMF in the SMF Type 70(series) records.

There are two variations in how SMF and RMF write interval data: (1) non-synchronized and (2) synchronized. Synchronization of SMF and RMF records is an option that must be explicitly specified in the SMFPRMxx member of SYS1.PARMLIB.

- **Non-synchronized writing of SMF and RMF interval data.** With non-synchronized writing of SMF and RMF interval data, the Sub-type 2 records are written based upon the interval lapse from the start of the job step. They are not written at the same time as is the SMF Type 70(Series) records. This lack of coordination between recording the two record types poses a correlation problem: a particular Sub-type 2 (or Sub-type 3) record may span between two RMF recording intervals. From a data analysis view, there is no way to precisely determine whether the data reflected in the Sub-type 2 record (or Sub-type 3 record) should belong to the first RMF Type 70(series) interval or should belong to the second RMF Type 70(series) interval.

  For example, suppose that the RMF recording interval were specified as 30 minutes, and RMF was directed to synchronize on the hour and half-hour. The RMF data would be collected and recorded at 10:00, 10:30, 11:00, 11:30, etc. Further suppose that a particular job step started at 15 minutes past the hour. Assuming that the Type 30 interval recording were specified as 30 minutes, SMF would create a Type 30 (Sub-type 2) interval record at 45 minutes past the hour, 15 past the next hour, and so forth. Thus, the RMF data would be recorded on the hour and half-hour, while the Sub-type 2 data would be recorded "offset" by 15 minutes.

  CPExpert addresses this problem by pro-rating the SMF Type 30 information based upon elapsed time. In the above example, 50% of the actual workload data contained in the SMF Type 30 (Sub-type 2 or Sub-type 3) records would be attributed to one RMF measurement interval and 50% would be attributed to the next RMF measurement interval. This pro-rating approach essentially assumes that the resources required by a job step do not vary much from one instant to the next.

  This approach works quite nicely so long as the job step uses resources in a uniform fashion. Many job steps exhibit this characteristic, and the resources required by the job step do not vary much as the job executes. Resources distributed using the pro-rating approach result in fairly consistent usage characteristics when comparing the summarized Type 30 data with SMF Type 70(series) data.

However, some job steps exhibit significant cycles, or require resources at the beginning or end of the job step. For these job steps, the pro-rating approach does not properly distribute the resource usage into the correct RMF measurement interval. Summarized Type 30 data would not compare well with Type 70(series) data if many job steps exhibit this cyclic or burst nature of resource usage. Unfortunately, there is no way to better distribute the data. Consequently, analysis based upon Type 30 data must be viewed with some caution. The analysis **generally** will be sufficiently precise for performance analysis purposes. However, anomalies will appear and results must always be subjected to a "reality" test.

This point is significant for the DASD Component, because the DASD Component attributes DASD usage to workloads based upon correlating SMF Type 30 data with SMF Type 74 data. The DASD I/O activity at the job step level is obtained from the SMF Type 30 interval records (using a modification to MXG or to MICS). This DASD I/O activity is pro-rated to the RMF measurement intervals as described above. The RMF DASD device I/O characteristics (IOSQ, PEND, DISC, and CONN times) are attributed to workloads based upon the pro-rating.

It is possible that the pro-rating method will result in improper attribution of I/O device characteristics to workloads. For example, suppose that a job step completed a few minutes past the hour (and that RMF data records were synchronized on the hour and half-hour). When the job step completed, it could execute many DASD I/O operations. These I/O operations would mostly be attributed to the previous RMF interval and the DASD device characteristics of that interval would be associated with most of the I/O operations. Suppose that there were no I/O problems with the device in the first interval. It is possible, however, that as the job step completed, it experienced significant DASD I/O problems which would be reflected in the second RMF interval. Since only a few of its I/O operations would be attributed to the second RMF interval, CPExpert would associate the I/O problems to only a few of its I/O operations. Consequently, CPExpert might consider the job step (and the workload category associated with the job step) to receive good DASD service, when the workload actually received bad service because of its burst I/O operations.

This example is not intended to invalidate the techniques CPExpert uses. Rather, the example is presented to explain a unique situation in which the techniques could result in improper conclusions. Readers may note that IBM's Service Level Reporter and any other software analyzing SMF/RMF data face the same analysis problem. The problem is with the data; not with the technique.

- **Synchronized writing of SMF and RMF interval data.** Synchronized writing of SMF and RMF was introduced with MVS/ESA SP4. When interval accounting is synchronized, SMF generates interval records for a work unit based on the end of the

SMF global recording interval, rather than the start time of a job. This feature allows Type 30 records (and other record types) to be synchronized with writing RMF Type70(series) records. SMF places indicators (or “flags”) in SMF Type70(series) records to indicate whether SMF and RMF records are synchronized.

It is not necessary for CPExpert to pro-rate data if the recording intervals are synchronized.

Chapter 2.2: Data Averages

The data collected by RMF and recorded in SMF Type 70(series) records provide a valuable source of information about the use and interaction of system components and workloads. However, the data are summarized and recorded at specific intervals (e.g., every 30 minutes). For most data elements, analysis must be accomplished based upon the **summary** or **average** values.

For example, the DASD IOSQ time reported for each device is the total for the measurement interval. The average IOSQ time per I/O operation is computed by dividing the total IOSQ time by the number of I/O operations. This average may have no relation to the IOSQ time experienced by any particular I/O operation. This problem is particularly pervasive as the RMF recording interval becomes more lengthy (e.g., if the recording interval were 60 minutes). The DASD IOSQ time may be quite long during the first half of the RMF measurement interval when contending workloads execute. The DASD IOSQ time may be short during the last half when a workload executes without contention from other applications. The average of the two extremes may lead to a conclusion that there was no problem with DASD IOSQ time for the entire interval!

Most DASD analysis performed by CPExpert assumes either a uniform or an exponential distribution of DASD I/O operations. For example, the pro-rating discussed in the previous chapter assumes a uniform distribution of I/O operations on a **job step** basis, over the life of the job step. However, the queuing models employed by CPExpert to analyze various aspects of DASD delays generally assume an exponential distribution of I/O operations at the **device** level, over an entire RMF measurement interval. Neither of these assumptions may be correct.

Wicks[3] illustrates a variety of distributions of the arrival rate of events, ranging from uniform distribution, to "cafeteria" distribution (events mostly arrive in clusters), to "London bus" distribution (events arrive only in clusters), to a random distribution (events exhibit a Poison or exponential arrival). The arrival of many events in computer systems exhibit an exponential distribution, and M/M/1 or M/M/C queuing models can fairly represent many aspects of the systems.

[3] Wicks, R. J., "*Balanced Systems and Capacity Planning*, IBM Corporation Washington Systems Center Technical Bulletin GG22-9299-02

However, Wicks gives an excellent example of exceptions: when editing a dataset using ISPF, the entire dataset may be read, some time is spent editing, and then the entire dataset may be written. The I/O requests in this instance would be similar to Wicks' "London Bus" distribution.

There is a tradeoff between (1) recording RMF data frequently, incurring the overhead and storage requirements of the additional RMF records, and requiring additional resources to analyze the data versus (2) recording RMF data less frequently, having less precise or representative data to analyze, and minimizing the resources required to perform the analysis. The importance of these tradeoffs must be evaluated in light of the objectives of the analysis and the requirements for precision of results.

In any event, any review of analysis and conclusions (whether by CPExpert or by a performance analyst) must be viewed with some caution because of the data summary, data averaging, and data distribution issues.

If the analysis consistently results in the same conclusions, you can be reasonably sure that the analysis is correct. However, it generally is unwise to make changes based upon analysis of a single day's RMF measurement information unless a "reality test" indicates that the analysis clearly is correct.